Package ‘bmem’

February 19, 2015

Type Package
Title Mediation analysis with missing data using bootstrap
Version 1.5
Date 2011-01-04
Author Zhiyong Zhang and Lijuan Wang
Maintainer Zhiyong Zhang <zhiyongzhang@nd.edu>
Depends R (>= 1.7), Amelia, MASS, lavaan, sem, snowfall
Description Four methods for mediation analysis with missing data: Listwise deletion, Pairwise deletion, Multiple imputation, and Two Stage Maximum Likelihood algorithm. For MI and TS-ML, auxiliary variables can be included. Bootstrap confidence intervals for mediation effects are obtained. The robust method is also implemented for TS-ML. Since version 1.4, bmem adds the capability to conduct power analysis for mediation models.
License GPL-2
LazyLoad yes
URL http://nd.psychstat.org
ZipData no
NeedsCompilation no
Repository CRAN
Date/Publication 2013-10-08 08:18:14

R topics documented:

  bmem-package .................................................. 2
  bmem .......................................................... 3
  bmem.bs ....................................................... 4
  bmem.ci.bc .................................................... 5
  bmem.ci.bc1 ................................................... 6
  bmem.ci.bca ................................................... 6
  bmem.ci.bca1 .................................................. 7
  bmem.ci.norm ................................................... 7
  bmem.ci.p ...................................................... 8
### Description

Four methods for mediation analysis with missing data: Listwise deletion, Pairwise deletion, Multiple imputation, and Two-stage ML. For MI and TSML, auxiliary variables can be included. Bootstrap confidence intervals for mediation effects are obtained.

### Details
Author(s)

Zhiyong Zhang and Lijuan Wang
Maintainer: Zhiyong Zhang <zhiyongzhang@nd.edu>

Description

Mediation analysis based on bootstrap

Usage

bmem(x, ram, indirect, v, method='tsml', ci='bc', cl=.95,
    boot=1000, m=10, varphi=.1, st='i', robust=FALSE,
    max_it=500, moment=FALSE, ...)

Arguments

x A data set
ram RAM path for the mediation model
indirect A vector of indirect effect
v Indices of variables used in the mediation model. If omitted, all variables are used.
ci norm: normal approximation CI, perc: percentile CI, bc: bias-corrected CI, bca: BCa
cl Confidence level. Can be a vector.
boot Number of bootstraps
m Number of imputations
varphi Percent of data to be downweighted
Starting values

Robust method

Select mean structure or covariance analysis. moment=FALSE, covariance analysis. moment=TRUE, mean and covariance analysis.

Maximum number of iterations in EM

Other options for sem function can be used.

The indirect effect can be specified using equations such as a*b, a*b+c, and a*b*c+d*e+f. A vector of indirect effects can be used indirect=c('a*b', 'a*b+c').

The on-screen output includes the parameter estimates, bootstrap standard errors, and CIs.

Zhiyong Zhang and Lijuan Wang


Bootrap but using the Bollen-Stine method

The same as bmem but using the Bollen-Stine method

Arguments

x: A data set
ram: RAM path for the mediation model
indirect: A vector of indirect effects
v: Indices of variables used in the mediation model. If omitted, all variables are used.

norm: normal approximation CI, perc: percentile CI, bc: bias-corrected CI, bca: BCa
bmem.ci.bc

```r
bmem.ci.bc(par.boot, par0, cl=.95)
```

---

### Description

Bias-corrected confidence intervals

### Usage

```r
bmem.ci.bc(par.boot, par0, cl=.95)
```

### Arguments

- `par.boot` A bootstrap object.
- `par0` Original estimate
- `cl` Confidence level. Default 0.95.

### Value

BC confidence intervals. The output includes estimates, bootstrap standard errors, and confidence intervals.

### Author(s)

Zhiyong Zhang and Lijuan Wang
See Also

`bmem.ci.norm, bmem.ci.p, bmem.ci.bca`

---

**bmem.ci.bcl**

*Bias-corrected confidence intervals (for a single variable)*

**Description**

Bias-corrected confidence intervals (for a single variable)

**Usage**

`bmem.ci.bcl(x, b, cl = 0.95)`

**Arguments**

- `x`: A vector from a bootstrap output.
- `b`: Parameter estimate from the original sample
- `cl`: Confidence level. Default 0.95.

**Author(s)**

Zhiyong Zhang and Lijuan Wang

**See Also**

`bmem.ci.norm, bmem.ci.p, bmem.ci.bca`

---

**bmem.ci.bca**

*Bias-corrected and accelerated confidence intervals*

**Description**

Bias-corrected and accelerated confidence intervals

**Usage**

`bmem.ci.bca(par.boot, par0, jack, cl = 0.95)`

**Arguments**

- `par.boot`: A bootstrap object.
- `par0`: Original estimate
- `jack`: A Jackknife object.
- `cl`: Confidence level. Default 0.95.
### Value
BCa confidence intervals. The output includes - estimates, bootstrap standard errors, and confidence intervals.

### Author(s)
Zhiyong Zhang and Lijuan Wang

### See Also
- `bmem.ci.norm`, `bmem.ci.p`, `bmem.ci.bc`, `bmem.list.jack`, `bmem.pair.jack`, `bmem.mi.jack`, `bmem.em.jack`

---

#### bmem.ci.bca1

**BCa for a single variable**

### Description
BCa for a single variable

### Usage
```r
bmem.ci.bca1(x, b, jack, cl = 0.95)
```

### Arguments
- **x**: A vector from a bootstrap output.
- **b**: Parameter estimate from the original sample
- **jack**: A vector from a Jackknife analysis
- **cl**: Confidence level. Default 0.95.

---

#### bmem.ci.norm

**Confidence interval based on normal approximation**

### Description
Confidence interval based on normal approximation

### Usage
```r
bmem.ci.norm(par.boot, par0, cl = 0.95)
```

### Arguments
- **par.boot**: A bootstrap object.
- **par0**: Original estimate
- **cl**: Confidence level. Default 0.95.
Value

Normal confidence intervals. The output includes estimates, bootstrap standard errors, and confidence intervals.

Author(s)

Zhiyong Zhang and Lijuan Wang

See Also

bmem.ci.bca, bmem.ci.p, bmem.ci.bc

Description

Percentile confidence interval

Usage

bmem.ci.p(par.boot, par0, cl = 0.95)

Arguments

par.boot A bootstrap object.
par0 Original estimate
cl Confidence level. Default 0.95.

Value

Percentile confidence intervals. The output includes estimates, bootstrap standard errors, and confidence intervals.

Author(s)

Zhiyong Zhang and Lijuan Wang

See Also

bmem.ci.bca, bmem.ci.norm, bmem.ci.bc
bmem.cov

*Calculate the covariance matrix based on a given ram model*

**Description**

Can be used to simulated data for an SEM model.

**Usage**

```r
bmem.cov(ram, obs.variables, moment=FALSE, debug=FALSE)
```

**Arguments**

- **ram**: An ram model
- **obs.variables**: Names of the observed variables
- **moment**: Whether to use the mean structure
- **debug**: Debug mode

bmem.em

*Estimate a mediation model based on EM covariance matrix*

**Description**

Estimate a mediation model based on EM covariance matrix

**Usage**

```r
bmem.em(x, ram, indirect, v, robust = FALSE, varphi = 0.1, st = "i", moment = FALSE, max_it = 500, ...)
```

**Arguments**

- **x**: A data set
- **ram**: RAM path for the mediation model
- **indirect**: A vector of indirect effects
- **v**: Indices of variables used in the mediation model. If omitted, all variables are used.
- **robust**: Robust method
- **varphi**: Percent of data to be downweighted
- **st**: Starting values
- **moment**: Select mean structure or covariance analysis. moment=FALSE, covariance analysis. moment=TRUE, mean and covariance analysis.
- **max_it**: Maximum number of iterations in EM
- **...**: Other options for `sem` function can be used.
**bmem.em.boot**  
*Bootstrap for EM*

**Description**
Bootstrap for EM

**Usage**

```r
bmem.em.boot(x, ram, indirect, v, robust = FALSE,
              varphi = 0.1, st = "i", boot = 1000,
              moment = FALSE, max_it = 500, ...)
```

**Arguments**
- **x**: A data set
- **ram**: RAM path for the mediation model
- **indirect**: A vector of indirect effects
- **v**: Indices of variables used in the mediation model. If omitted, all variables are used.
- **robust**: Robust method
- **varphi**: Percent of data to be downweighted
- **st**: Starting values
- **boot**: Number of bootstraps. Default is 1000.
- **moment**: Select mean structure or covariance analysis. `moment=FALSE`, covariance analysis. `moment=TRUE`, mean and covariance analysis.
- **max_it**: Maximum number of iterations in EM
- **...**: Other options for `sem` function can be used.

**Details**
The indirect effect can be specified using equations such as \(ab\), \(ab+c\), and \(ab+cd+e+f\). A vector of indirect effects can be used `indirect=c('ab', 'ab+c')`.

**Value**
- **par.boot**: Parameter estimates from bootstrap samples
- **par0**: Parameter estimates from the original samples

**Author(s)**
Zhiyong Zhang and Lijuan Wang
**bmem.em.cov**
*Covariance matrix from EM*

**Description**
Covariance matrix from EM

**Usage**
```
bmem.em.cov(xmis, moment = FALSE, max_it = 500)
```

**Arguments**
- **xmis**: An object from output of `bmem.pattern`.
- **moment**: Whether estimating mean
- **max_it**: Maximum number of iterations

---

**bmem.em.jack**
*Jackknife estimate using EM*

**Description**
Jackknife estimate using EM

**Usage**
```
bmem.em.jack(x, ram, indirect, v, robust = FALSE,
              varphi = 0.1, st = “i”, moment = FALSE,
              max_it = 500, ...)
```

**Arguments**
- **x**: A data set
- **ram**: RAM path for the mediation model
- **indirect**: A vector of indirect effect
- **v**: Indices of variables used in the mediation model. If omitted, all variables are used.
- **robust**: Robust method
- **varphi**: Percent of data to be downweighted
- **st**: Starting values
- **moment**: Select mean structure or covariance analysis. moment=FALSE, covariance analysis. moment=TRUE, mean and covariance analysis.
- **max_it**: Maximum number of iterations in EM
- **...**: Other options for `sem` function can be used.
bmem.em.rcov  

Estimation of robust covariance matrix

Description

Estimation of robust covariance matrix

Usage

bmem.em.rcov(xmis, varphi=.1, moment=FALSE, max_it=1000, st='i')

Arguments

- **xmis**: Missing data pattern
- **varphi**: Percent of data to be downweighted
- **moment**: Moment analysis if TRUE
- **max_it**: Maximum number of iteration
- **st**: Starting values

Value

An interval function to calculate the robust covariance matrix

Author(s)

Zhiyong Zhang and Lijuan Wang

bmem.list  

Estimate a mediation model based on listwise deletion

Description

Estimate a mediation model based on listwise deletion

Usage

bmem.list(x, ram, indirect, moment = FALSE, ...)

Arguments

- **x**: A data set
- **ram**: RAM path for the mediation model
- **indirect**: A vector of indirect effect
- **moment**: Select mean structure or covariance analysis. moment=FALSE, covariance analysis. moment=TRUE, mean and covariance analysis.
- **...**: Other options for `sem` function can be used.
**bmem.list.boot**  
*Bootstrap for listwise deletion method*

**Description**

Bootstrap for listwise deletion method

**Usage**

```r
bmem.list.boot(x, ram, indirect, boot = 1000, moment = FALSE, ...)
```

**Arguments**

- `x`: A data set
- `ram`: RAM path for the mediation model
- `indirect`: A vector of indirect effect
- `boot`: Number of bootstraps. Default is 1000.
- `moment`: Select mean structure or covariance analysis. `moment=FALSE`, covariance analysis. `moment=TRUE`, mean and covariance analysis.
- `...`: Other options for `sem` function can be used.

**bmem.list.cov**  
*Covariance matrix for listwise deletion*

**Description**

Covariance matrix for listwise deletion

**Usage**

```r
bmem.list.cov(x, moment = FALSE)
```

**Arguments**

- `x`: A data set
- `moment`: Estimate mean or not
bmem.list.jack \hspace{1cm} \textit{Jackknife for listwise deletion}

**Description**

Jackknife for listwise deletion

**Usage**

\texttt{bmem.list.jack(x, ram, indirect, moment = FALSE, \ldots)}

**Arguments**

- \texttt{x} \hspace{0.5cm} A data set
- \texttt{ram} \hspace{0.5cm} RAM path for the mediation model
- \texttt{indirect} \hspace{0.5cm} A vector of indirect effect
- \texttt{moment} \hspace{0.5cm} Select mean structure or covariance analysis. \texttt{moment=FALSE}, covariance analysis. \texttt{moment=TRUE}, mean and covariance analysis.
- \texttt{\ldots} \hspace{0.5cm} Other options for \texttt{sem} function can be used.

bmem.mi \hspace{1cm} \textit{Estimate a mediation model based on multiple imputation}

**Description**

Estimate a mediation model based on multiple imputation

**Usage**

\texttt{bmem.mi(x, ram, indirect, v, m = 10, moment = FALSE, \ldots)}

**Arguments**

- \texttt{x} \hspace{0.5cm} A data set
- \texttt{ram} \hspace{0.5cm} RAM path for the mediation model
- \texttt{indirect} \hspace{0.5cm} A vector of indirect effect
- \texttt{v} \hspace{0.5cm} Indices of variables used in the mediation model. If omitted, all variables are used.
- \texttt{m} \hspace{0.5cm} Number of imputations.
- \texttt{moment} \hspace{0.5cm} Select mean structure or covariance analysis. \texttt{moment=FALSE}, covariance analysis. \texttt{moment=TRUE}, mean and covariance analysis.
- \texttt{\ldots} \hspace{0.5cm} Other options for \texttt{sem} function can be used.
bmem.mi.boot  

**Bootstrap for multiple imputation**

**Description**

Bootstrap for multiple imputation

**Usage**

```r
bmem.mi.boot(x, ram, indirect, v, m = 10, boot = 1000, moment = FALSE, ...)
```

**Arguments**

- `x`: A data set
- `ram`: RAM path for the mediation model
- `indirect`: A vector of indirect effects
- `v`: Indices of variables used in the mediation model. If omitted, all variables are used.
- `m`: Number of imputations
- `boot`: Number of bootstraps. Default is 1000.
- `moment`: Select mean structure or covariance analysis. `moment=FALSE`, covariance analysis. `moment=TRUE`, mean and covariance analysis.
- `...`: Other options for `sem` function can be used.

bmem.mi.cov  

**Covariance estimation for multiple imputation**

**Description**

Covariance estimation for multiple imputation

**Usage**

```r
bmem.mi.cov(x, m = 10, moment = FALSE)
```

**Arguments**

- `x`: A data set
- `m`: Number of imputations
- `moment`: Estimate mean or not
bmem.moments

Jackknife for multiple imputation

Description

Jackknife for multiple imputation

Usage

bmem.mi.jack(x, ram, indirect, v, m = 10, moment = FALSE, ...)

Arguments

x         A data set
ram       RAM path for the mediation model
indirect  A vector of indirect effects
v         Indices of variables used in the mediation model. If omitted, all variables are used.
m         Number of imputations.
moment    Select mean structure or covariance analysis. moment=FALSE, covariance analysis. moment=TRUE, mean and covariance analysis.
...       Other options for sem function can be used.

bmem.moments

Calculate the moments of a data set

Description

Calculate the moments of a data set using either listwise deletion or pairwise deletion

Usage

bmem.moments(x, type=0)

Arguments

x         A data set
type      How to deal with missing data. 0: listwise deletion; 1: pairwise deletion
**bmem.pair**

Estimate a mediation model based on pairwise deletion

**Description**

Estimate a mediation model based on pairwise deletion

**Usage**

```r
bmem.pair(x, ram, indirect, moment = FALSE, ...)
```

**Arguments**

- `x`: A data set
- `ram`: RAM path for the mediation model
- `indirect`: A vector of indirect effects
- `moment`: Select mean structure or covariance analysis. `moment=FALSE`, covariance analysis. `moment=TRUE`, mean and covariance analysis.
- `...`: Other options for `sem` function can be used.

---

**bmem.pair.boot**

Bootstrap for pairwise deletion

**Description**

Bootstrap for pairwise deletion

**Usage**

```r
bmem.pair.boot(x, ram, indirect, boot = 1000, moment = FALSE, ...)
```

**Arguments**

- `x`: A data set
- `ram`: RAM path for the mediation model
- `indirect`: A vector of indirect effects
- `boot`: Number of bootstraps. Default is 1000.
- `moment`: Select mean structure or covariance analysis. `moment=FALSE`, covariance analysis. `moment=TRUE`, mean and covariance analysis.
- `...`: Other options for `sem` function can be used.
Covariance matrix estimation based on pairwise deletion

Description
Covariance matrix estimation based on pairwise deletion

Usage
bmem.pair.cov(x, moment = FALSE)

Arguments
- x: A data set
- moment: Estimate mean or not

Jackknife for pairwise deletion

Description
Jackknife for pairwise deletion

Usage
bmem.pair.jack(x, ram, indirect, moment = FALSE, ...)

Arguments
- x: A data set
- ram: RAM path for the mediation model
- indirect: A vector of indirect effects
- moment: Select mean structure or covariance analysis. moment=FALSE, covariance analysis. moment=TRUE, mean and covariance analysis.
- ... Other options for sem function can be used.
**bmem.pattern**

*Obtain missing data pattern information*

**Description**

Obtain missing data pattern information

**Usage**

```r
bmem.pattern(x)
```

**Arguments**

- `x` A data set

**bmem.plot**

*Plot of the bootstrap distribution. This function is replaced by plot.*

**Description**

Plot of the bootstrap distribution

**Usage**

```r
bmem.plot(x, par,...)
```

**Arguments**

- `x` A bmem object
- `par` Name of parameter to be plotted.
- `...` Options used for the generic plot function.

**Value**

A plot

**Author(s)**

Zhiyong Zhang and Lijuan Wang

**References**

Zhang, Z. (2011) Robust mediation analysis with missing data and auxiliary variables.

**See Also**

`bmem`, `bmem.sobel`, `bmem.plot`
bmem.raw2cov

Convert a raw moment matrix to covariance matrix

Description
Convert a raw moment matrix to covariance matrix

Usage
bmem.raw2cov(x)

Arguments
x
A moment matrix

Value
A covariance matrix

Author(s)
Zhiyong Zhang and Lijuan Wang

References
Zhang, Z. (2011) Robust mediation analysis with missing data and auxiliary variables.

See Also
bmem, bmem.sobel, bmem.plot

bmem.sem

Estimate a mediation model using SEM technique

Description
Estimate a mediation model using SEM technique

Usage
bmem.sem(x, ram, N, indirect, moment=FALSE, ...)

Arguments

- `x`: A covariance matrix
- `ram`: A path diagram from `specify.model`
- `N`: Sample size
- `indirect`: A vector of indirect effects
- `moment`: Whether mean structure is used. The default is `FALSE`
- `...`: Options that can be supplied to function `sem`.

See Also

- `bmem.list.cov`, `bmem.pair.cov`, `bmem.mi.cov`, `bmem.em.cov`

Description

Mediation analysis using sobel test (for complete data only)

Usage

```r
bmem.sobel(x, ram, indirect, moment=FALSE, ...)
```

Arguments

- `x`: A data set
- `ram`: RAM path for the mediation model
- `indirect`: A vector of indirect effects
- `moment`: Covariance or moment analysis
- `...`: Other options for `sem` function can be used.

Value

The on-screen output includes the parameter estimates and sobel standard errors.

Author(s)

Zhiyong Zhang and Lijuan Wang

References

Zhang, Z. (2011) Robust mediation analysis with missing data and auxiliary variables.

See Also

- `bmem`, `bmem.sobel`, `bmem.plot`
bmem.sobel.ind  
*Mediation analysis using sobel test for one indirect effect*

**Description**  
Internal function

**Usage**  
`bmem.sobel.ind(sem.object, ind)`

**Arguments**  
- `sem.object`  A sem object  
- `ind`  Indirect effect

**Value**  
Internal output

**Author(s)**  
Zhiyong Zhang and Lijuan Wang

**References**  
Zhang, Z. (2011) Robust mediation analysis with missing data and auxiliary variables.

**See Also**  
`bmem, bmem.sobel, bmem.plot`

---

bmem.ssq  
*Sum square of a matrix*

**Description**  
Sum square of a matrix

**Usage**  
`bmem.ssq(x)`

**Arguments**  
- `x`  A matrix
**bmem.v**

*Select data according to a vector of indices*

---

**Description**

Select data according to a vector of indices

**Usage**

```r
bmem.v(x, v, moment = FALSE)
```

**Arguments**

- `x`: A matrix
- `v`: A vector of indices
- `moment`: Covariance analysis or mean and covariance analysis

---

**plot.bmem**

*Plot of the bootstrap distribution*

---

**Description**

Plot of the bootstrap distribution

**Usage**

```r
## S3 method for class 'bmem'
plot(x, par, ...)
```

**Arguments**

- `x`: A `bmem` object
- `par`: Name of parameter to be plotted.
- `...`: Options used for the generic plot function.

**Value**

Generate the bootstrap histogram for a chosen parameter.

**Author(s)**

Zhiyong Zhang and Lijuan Wang
References

Zhang, Z. (2011) Robust mediation analysis with missing data and auxiliary variables.

See Also

bmem, bmem.sobel, bmem.plot

popPar

Get the population parameter values

Description

Get the population parameter values including both direct and indirect effects in a model

Usage

popPar(object)

Arguments

object A lavaan object

power.basic

Conducting power analysis based on Sobel test

Description

Different from power.boot, this function conduct power analysis based on the Sobel test.

Usage

power.basic(model, indirect = NULL, nobs, nrep = 1000, alpha = 0.95,
            skewness = NULL, kurtosis = NULL, ovnames = NULL, se = "default",
estimator = "default", parallel = "no", ncore = 1, ...)
**Arguments**

*model*  
A model specified using lavaan notation and above. See `model.syntax` for basic model specification.

For the power analysis, the population parameter values should be provided in the following way. For example, the coefficient between math and HE is .39. Then it is specified as `start(.39)`. If the parameter will be referred in the mediation effect, a label should be given as a modifier as `b*HE+start(.39)*HE`.

```
model<-' math ~ c*ME+start(0)*ME + b*HE+start(.39)*HE HE ~ a*ME+start(.39)*ME`
```

*indirect*  
The indirect or other composite effects are specified in the following way

```
indirect<-' ab: = a*b abc := a*b + c`
```

*nobs*  
Number of observations for power analysis. If it is a vector, multiple group analysis will be conducted.

*nrep*  
Number of replications for Monte Carlo simulation. At least 1,000 is recommended.

*alpha*  
The alpha level is used to obtain the confidence interval for model parameters.

*skewness*  
A vector to give the skewness for the observed variables.

*kurtosis*  
A vector to give the kurtosis for the observed variables.

*ovnames*  
A vector to give the variable names for the observed variables. This is only needed when the skewness and kurtosis are provided. The skewness, kurtosis and variable names should be in the same order.

*se*  
How to calculate the standard error, for example, robust standard error can be specified using `se="robust"`.

*estimator*  
Estimation methods to be used here.

*parallel*  
Parallel methods, snow or multicore, can be used here.

*ncore*  
Number of cores to be used in parallel. By default, the maximum number of cores are used.

*...*  
Other named arguments for lavaan can be passed here.

**Value**

*power*  
power for all parameters and required ones in the model

*coverage*  
coverage probability

*pop.value*  
Population parameter values

*results*  
A list to give all intermediate results

*data*  
The last data set generated for checking purpose

**Examples**

```
# Not run:
ex1model<-' 
math ~ c*ME+start(0)*ME + b*HE+start(0.39)*HE 
HE ~ a*ME+start(0.39)*ME
```
Conducting power analysis based on bootstrap

**Description**

Different from `power.basic`, this function conduct power analysis based on the bootstrap method.

**Usage**

```r
power.boot(model, indirect = NULL, nobs, nrep = 1000, nboot = 1000, alpha = 0.95, skewness = NULL, kurtosis = NULL, ovnames = NULL, ci='default', boot.type='default', se = "default", estimator = "default", parallel = "no", ncore = 1, ...)```

**Arguments**

- `model` A model specified using lavaan notation and above. See `model.syntax` for basic model specification.
  
  For the power analysis, the population parameter values should be provided in the following way. For example, the coefficient between math and HE is .39. Then it is specified as start(.39). If the parameter will be referred in the mediation effect, a label should be given as a modifier as b*HE+start(.39)*HE.
  
  ```r
  model<-' math ~ c*ME+start(0)*ME + b*HE+start(.39)*HE HE ~ a*ME+start(.39)*ME '```

- `indirect` The indirect or other composite effects are specified in the following way
  
  ```r
  indirect<-' ab:=a*b '```

- `nobs` Number of observations for power analysis. If it is a vector, multiple group analysis will be conducted.

- `nrep` Number of replications for Monte Carlo simulation. At least 1,000 is recommended.

- `nboot` Number of bootstraps to conduct.
alpha The alpha level is used to obtain the confidence interval for model parameters.
skewness A vector to give the skewness for the observed variables.
kurtosis A vector to give the kurtosis for the observed variables.
ovnames A vector to give the variable names for the observed variables. This is only needed when the skewness and kurtosis are provided. The skewness, kurtosis and variable names should be in the same order.
se How to calculate the standard error, for example, robust standard error can be specified using se="robust".
estimator Estimation methods to be used here.
parallel Parallel methods, snow or multicore, can be used here.
ncore Number of cores to be used in parallel. By default, the maximum number of cores are used.
ci Type of bootstrap confidence intervals. By default, the percentile one is used. To get the bias-corrected one, use ci='BC'.
boot.type Type of bootstrap method. By default, the nonparametric one is used. Changing it to "BS" to use the Bollen-Stine method.
... Other named arguments for lavaan can be passed here.

Value

power power for all parameters and required ones in the model
coverage coverage probability
pop.value Population parameter values
results A list to give all intermediate results
data The last data set generated for checking purpose

Examples

## Not run:
ex1model<-
math ~ c*ME+start(0)*ME + b*HE+start(0.39)*HE
HE ~ a*ME+start(0.39)*ME
'
indirect<-'ab=a*b'

N<50

system.time(boot.non.normal<-power.boot(ex1model, indirect, N,
nrep=2000, nboot=10000, parallel='multicore', skewness=c(-.3, -.7, 1.3), kurtosis=c(1.5, 0, 5), ovnames=c('ME', 'HE', 'math'), ncore=8, ci='percent', boot.type='simple'))
summary(boot.non.normal)

## End(Not run)
Description

Generate a power curve either based on Sobel test or bootstrap

Usage

```r
power.curve(model, indirect=NULL, nobs=100, type='basic', nrep=1000,
nboot=1000, alpha=.95, skewness=NULL, kurtosis=NULL, ovnames=NULL,
ci='default', boot.type='default',
se="default", estimator="default", parallel="no",
ncore=1, interactive=TRUE, ...)
```

Arguments

- `model` A model specified using lavaan notation and above. See `model.syntax` for basic model specification.
  For the power analysis, the population parameter values should be provided in the following way. For example, the coefficient between math and HE is .39. Then it is specified as `start(.39)`. If the parameter will be referred in the mediation effect, a label should be given as a modifier as `b*HE+start(.39)*HE`.
  ```r
  model<-' math ~ c*ME+start(0)*ME + b*HE+start(.39)*HE
  HE ~ a*ME+start(.39)*ME
  '
  ```

- `indirect` The indirect or other composite effects are specified in the following way
  ```r
  indirect<-' ab: = a*b abc := a*b + c '
  ```

- `nobs` Number of observations for power analysis. It is typically should be a vector for single group analysis. For multiple group analysis, it should be a matrix.

- `type` Type of power analysis

- `nrep` Number of replications for Monte Carlo simulation. At least 1,000 is recommended.

- `nboot` Number of bootstraps to conduct.

- `alpha` The alpha level is used to obtain the confidence interval for model parameters.

- `skewness` A vector to give the skewness for the observed variables.

- `kurtosis` A vector to give the kurtosis for the observed variables.

- `ovnames` A vector to give the variable names for the observed variables. This is only needed when the skewness and kurtosis are provided. The skewness, kurtosis and variable names should be in the same order.

- `se` How to calculate the standard error, for example, robust standard error can be specified using `se="robust"`.

- `estimator` Estimation methods to be used here.

- `parallel` Parallel methods, snow or multicore, can be used here.
**ncore**  
Number of cores to be used in parallel. By default, the maximum number of cores are used.

**ci**  
Type of bootstrap confidence intervals. By default, the percentile one is used.  
To get the bias-corrected one, use *ci*='BC'

**boot.type**  
Type of bootstrap method. By default, the nonparametric one is used. Changing it to "BS" to use the Bollen-Stine method.

**interactive**  
Whether to get the figure interactively.

**...**  
Other named arguments for lavaan can be passed here.

---

**Value**

- **power**  
  power for all parameters and required ones in the model

- **coverage**  
  coverage probability

- **pop.value**  
  Population parameter values

- **results**  
  A list to give all intermediate results

- **data**  
  The last data set generated for checking purpose

---

**Examples**

```r
## Not run:
ex2model<-'
  ept ~ start(.4)*hvltt + b*hvltt + start(0)*age + start(0)*edu + start(2)*R
  hvltt ~ start(-.35)*age + a*age + c*edu + start(.5)*edu
  R ~ start(-.06)*age + start(.2)*edu
  R =~ 1*ws + start(.8)*ls + start(.5)*lt
  age ~~ start(30)*age
  edu ~~ start(8)*edu
  age ~~ start(-.8)*edu
  hvltt ~~ start(23)*hvltt
  R ~~ start(14)*R
  ws ~~ start(3)*ws
  ls ~~ start(3)*ls
  lt ~~ start(3)*lt
  ept ~~ start(3)*ept

indirect<-'ind1 := a*b + c*b'

nobs <- seq(100, 2000, by=200)

power.curve(model = ex2model, indirect = indirect, nobs = nobs,
    type = 'boot', parallel = 'multicore', ncore = 60, ci = 'percent',
    boot.type = 'simple', interactive = f)

## End(Not run)
```
### summary.bmem

**Calculate bootstrap confidence intervals**

**Description**

Calculate bootstrap confidence intervals

**Usage**

```r
## S3 method for class 'bmem'
summary(object, ci='bc', cl=.95, ...)
```

**Arguments**

- **object**: An output object from the function `bmem`
- **ci**: `norm`: normal approximation CI, `perc`: percentile CI, `bc`: bias-corrected CI, `bca`: BCa
- **cl**: Confidence level. Can be a vector.
- **...**: Other options can be used for the generic `summary` function.

**Details**

The other type of confidence intervals can be constructed from the output of the function `bmem`. Note if the BCa is required, the `ci='Bca'` should have been specified in the function `bmem`.

**Value**

The on-screen output includes the parameter estimates, bootstrap standard errors, and CIs.

### summary.power

**Organize the results into a table**

**Description**

This function is adapted from the `lavaan` summary function to put the results in a table.

**Usage**

```r
## S3 method for class 'power'
summary(object,...)
```

**Arguments**

- **object**: Output from the function either `power.basic` or `power.boot`.
- **...**: Other options
Index

*Topic bmem

bmem, 3, 4, 5, 19–22, 24, 30
bmem-package, 2
bmem.bs, 4
bmem.ci.bc, 5, 7, 8
bmem.ci.bcl, 6
bmem.ci.bca, 6, 6, 8
bmem.ci.bcal, 7
bmem.ci.norm, 6, 7, 7, 8
bmem.ci.p, 6–8, 8
bmem.cov, 9
bmem.em, 9
bmem.em.boot, 10
bmem.em.cov, 11, 21
bmem.em.jack, 7, 11
bmem.em.rcov, 12
bmem.list, 12
bmem.list.boot, 13
bmem.list.cov, 13, 21
bmem.list.jack, 7, 14
bmem.mi, 14
bmem.mi.boot, 15
bmem.mi.cov, 15, 21
bmem.mi.jack, 7, 16
bmem.moments, 16
bmem.pair, 17
bmem.pair.boot, 17
bmem.pair.cov, 18, 21
bmem.pair.jack, 7, 18
bmem.pattern, 11, 19
bmem.plot, 5, 19, 19, 20–22, 24
bmem.raw2cov, 20
bmem.sem, 20
bmem.sobel, 5, 19–21, 21, 22, 24
bmem.sobel.ind, 22
bmem.ssq, 22
bmem.v, 23

lavaan, 24, 30

model.syntax, 25, 26, 28
plot(plot.bmem), 23
plot.bmem, 23
popPar, 24
power.basic, 24, 26, 30
power.boot, 24, 26, 30
power.curve, 28

sem, 4, 5, 9–18, 21
summary(summary.bmem), 30
summary.bmem, 30
summary.power, 30